

ssh – The Secure Shell

David Morgan

© David Morgan 2003

A client-server pair of programs

- ssh - client
 - /usr/bin/ssh (configurable)
- sshd - server
 - /usr/sbin/sshd
 - assigned port number 22
- Originated by TatuYlonen as a secure drop-in replacement for rsh/rlogin/rcp

© David Morgan 2003

Secure Shell's Functions

- Explicit functionalities
 - Remote login tool
 - Remote command executor
- Implicit activities
 - Authentication
 - Encryption

© David Morgan 2003

ssh stated mission

“ssh is a program for *logging into* a remote machine and for *executing commands* in a remote machine. It is intended to replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure channel.”

ssh man page – first sentence

© David Morgan 2003

ssh syntax

Logging in

```
ssh -l remote-user-name remote-machine-id
```

e.g., `ssh -l root 193.6.37.12`

Executing a command

```
ssh -l remote-user-name remote-machine-id command
```

e.g., `ssh -l root 193.6.37.12 cat /etc/passwd`

© David Morgan 2003

ssh dynamic encryption

- all session/command traffic passes through ssh/sshd (sshd runs on port 22)
- encrypted going out/decrypted coming in
- for duration of session/command

© David Morgan 2003

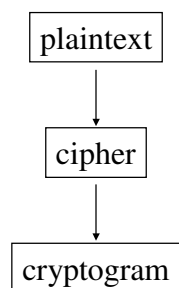
ssh – why secure?

- uses RSA (public-key) authentication
- then strong-key symmetrical encryption

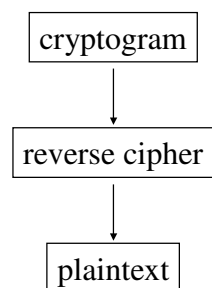
© David Morgan 2003

Cryptographic processing

Encryption

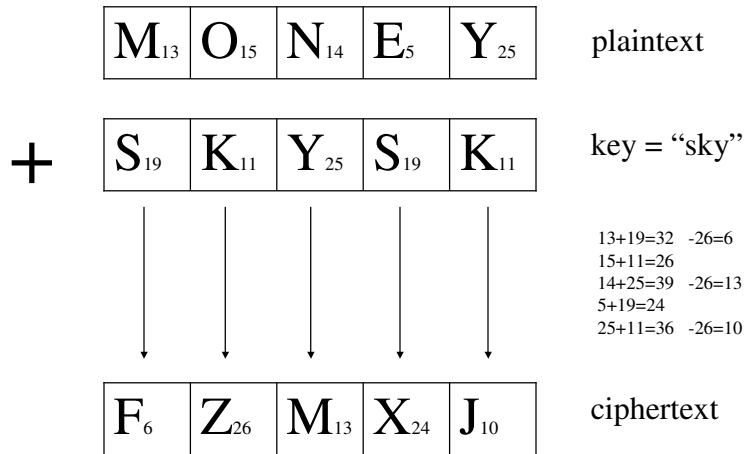


Decryption



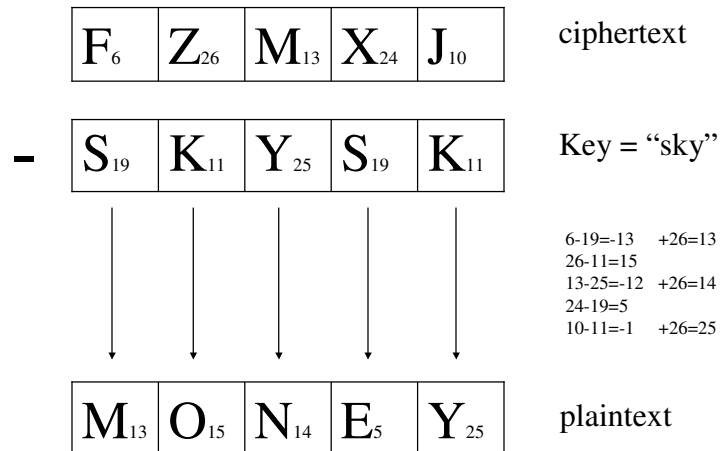
© David Morgan 2003

Key cipher - encryption



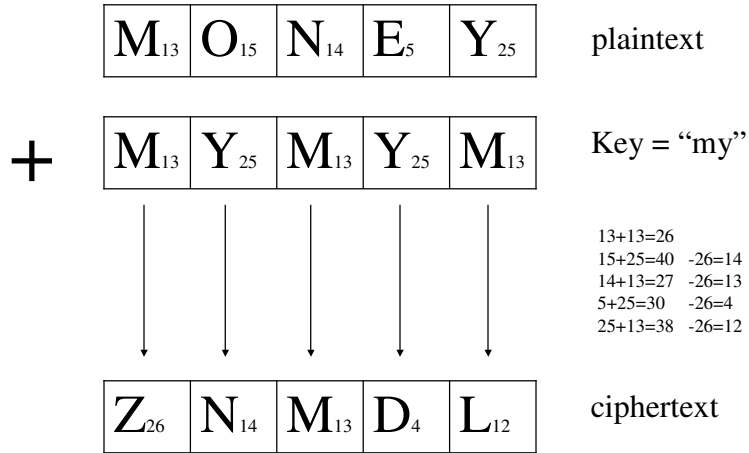
© David Morgan 2003

Key cipher - decryption



© David Morgan 2003

Ciphertext is key-dependent



13+13=26
15+25=40 -26=14
14+13=27 -26=13
5+25=30 -26=4
25+13=38 -26=12

not FZMXJ

© David Morgan 2003

Cryptography systems: issues

- Number of keys?
- Locus of decrypt key origination?
- Locus of decrypt key utilization?
- Does decrypt key have to travel?
- Is there in-transit interception risk?

© David Morgan 2003

Secret (single)-key Cryptography

- One key - same key encrypts *and* decrypts
- Decrypt key origination: encryptor's place
- Decrypt key utilization: decryptor's place
- Different places
- Decrypt key *must travel*

© David Morgan 2003

Public (dual)-key Cryptography

- Two keys - one encrypts, other decrypts
(keys are mathematically paired)
- Decrypt key origination: decryptor's place
- Decrypt key utilization: decryptor's place
- Same place
- Decrypt key *never travels*

© David Morgan 2003

ssh implementation

When local machine issues:

```
ssh -l remote-user-name remote-machine-id
```

Local machine (ssh) sends *local* user's public key to remote machine (sshd)

Remote machine authenticates if 1) that key appears in remote-user-name's authorized_keys file, if so 2) local machine can decrypt random text encrypted with it by remote machine as a challenge.

© David Morgan 2003

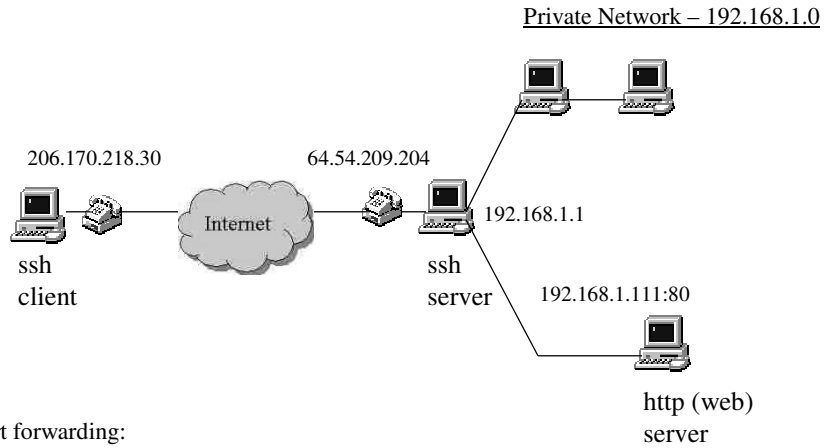
Configuration files

- ssh - /etc/ssh/ssh_config
 - Cipher selection
 - Compression level
 - Port forwardings

- sshd - /etc/ssh/sshd_config
 - authentication options
 - Key-based
 - Password
 - both
 - Logging options

© David Morgan 2003

ssh feature: port forwarding



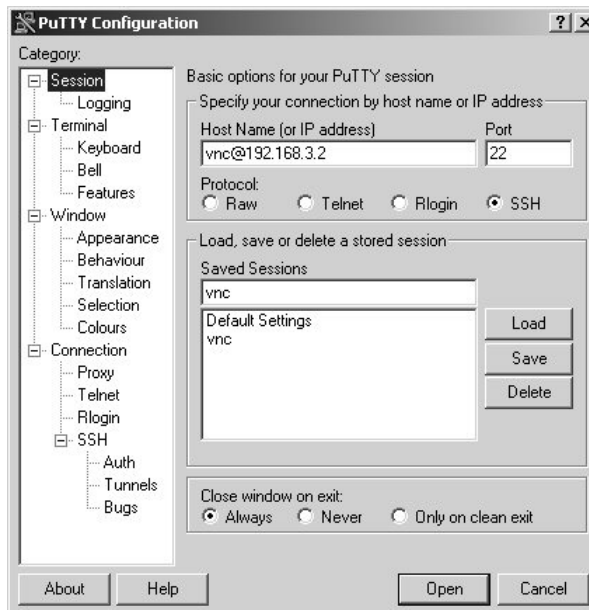
ssh port forwarding:

correspond some port on the client (e.g., 3000) to
some port (e.g., 80) on a machine reachable thru the server....

Example: `http://127.0.0.1:3000` in client's browser gets served from 192.168.1.111

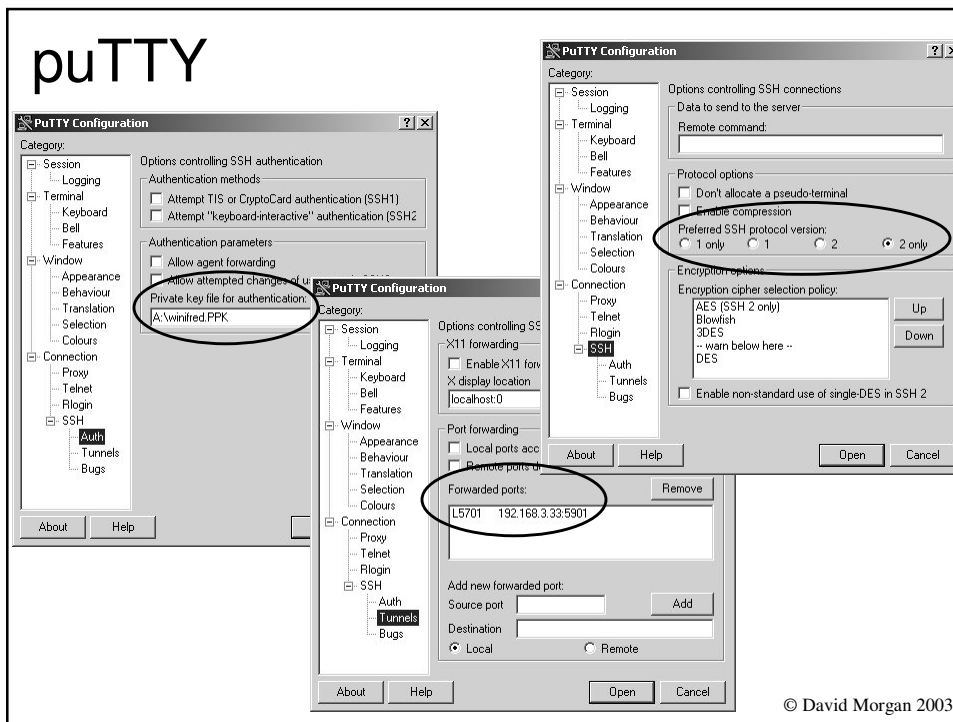
© David Morgan 2003

puTTY

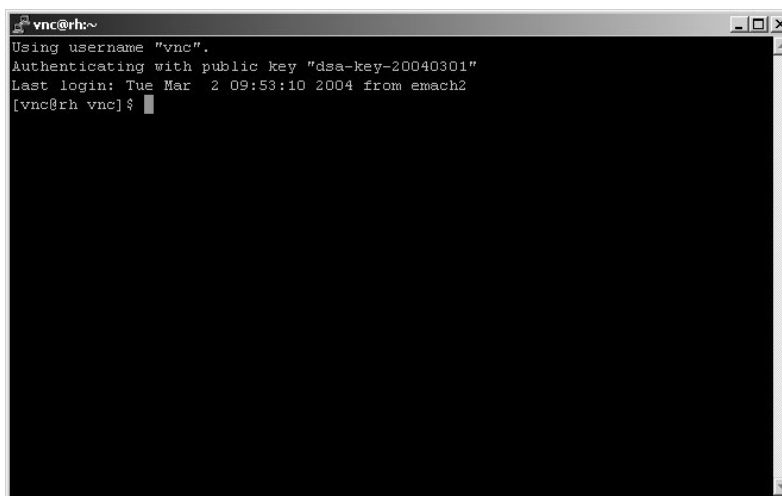


© David Morgan 2003

puTTY



puTTY



Don't leave back door ajar!

“System security is not improved unless rshd, rlogind, rexecd, and rexd are disabled (thus completely disabling rlogin and rsh into that machine).”

sshd man page

© David Morgan 2003

Getting ssh

- commercial
 - <http://www.ssh.com/>
 - <http://www.f-secure.com/>
- free
 - <http://www.openssh.com/>
 - (ssl is prerequisite <http://www.openssl.org/>)
- ftp site – encryption download
 - <http://www.zedz.net/>

© David Morgan 2003

Free windows clients

- puTTY

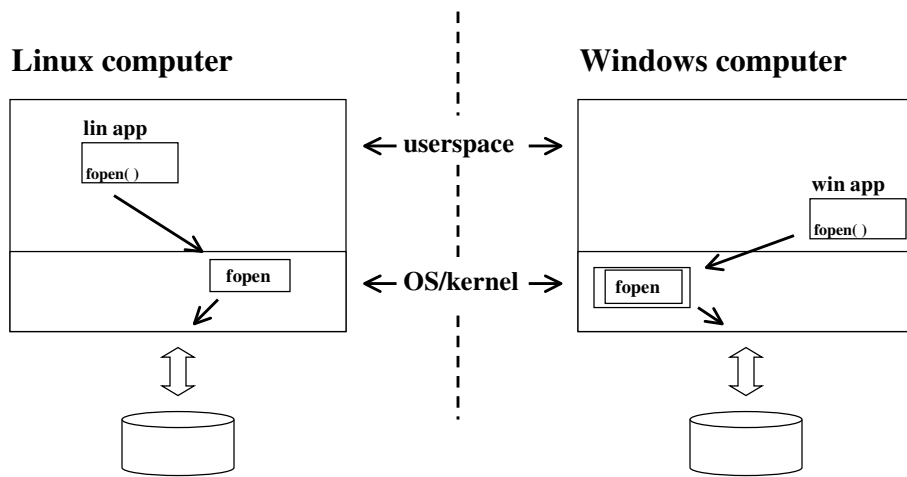
<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

- cygwin under Windows / openSSH under cygwin

<http://www.cygwin.com/>

© David Morgan 2003

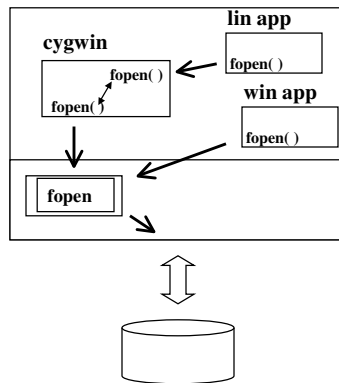
api for OS services, eg fopen()



© David Morgan 2003

api mapping by cygwin

Windows computer



- cygwin is general
- any lin app that's been adapted can run
- openSSH has been adapted
- openSSH can run under Windows

© David Morgan 2003

ssh information

- ssh FAQ
<http://www.ssh.org/>
- “Getting Started with ssh”
<http://kimmo.suominen.com/ssh/>
- ssh resource page
<http://www.csri.utoronto.ca/~djast/ssh.html>

© David Morgan 2003

Please don't tell...

... it's a secret.

© David Morgan 2003